

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.

**THIS PAGE BLANK (USPTO)**

# (12) UK Patent Application (19) GB (11) 2 231 419 (13) A

(43) Date of A publication 14.11.1990

(21) Application No 8910294.1

(22) Date of filing 05.05.1989

(71) Applicant  
Technophone Limited  
(Incorporated in the United Kingdom)

Ashwood House, Pembroke Broadway, Camberley,  
Surrey, GU15 3SP, United Kingdom

(72) Inventors  
Christopher Michael Harrison  
Mark Powell  
Anthony John Whitley  
Michael Overy

(74) Agent and/or Address for Service  
T J Frain  
Technophone Limited, Ashwood House,  
Pembroke Broadway, Camberley, Surrey, GU15 3SP,  
United Kingdom

(51) INT CL<sup>3</sup>  
G06F 11/14 9/06 9/44

(52) UK CL (Edition K)  
G4A AEF APX A12D

(56) Documents cited  
GB 2122780 A GB 2043308 A GB 1488271 A  
GB 1346219 A US 4095278 A

(58) Field of search  
UK CL (Edition J) G4A AEF AFMP APX  
INT CL<sup>4</sup> G06F  
On-line databases: WPI

## (54) Program-updating PROM

(57) A PROM (10) has a pre-programmed portion (10a) comprising one or more sub-routines (1-3) of instructional information, a look-up table (4), and an unprogrammed portion (10b). If a sub-routine (2) is found to contain a programming error, i.e. a bug (6), the whole of that sub-routine is re-written as a new sub-routine (2A) in the unprogrammed portion of the PROM. The addresses of the original and new sub-routines are entered into the look-up table in correlated relationship. An instruction is provided at the start of the defective sub-routine (2) which causes the look-up table to be consulted before executing the rest of the instructions in that sub-routine. The new sub-routine is executed in place of the original sub-routine if the look-up table contains information correlating an original sub-routine and a new sub-routine. If the look-up table does not contain correlation information the original sub-routine is executed.

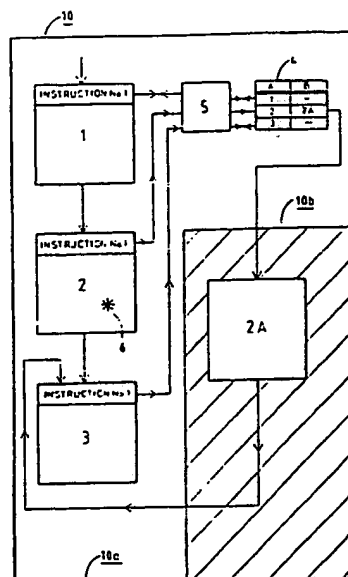


FIG 1

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

GB 2 231 419 A

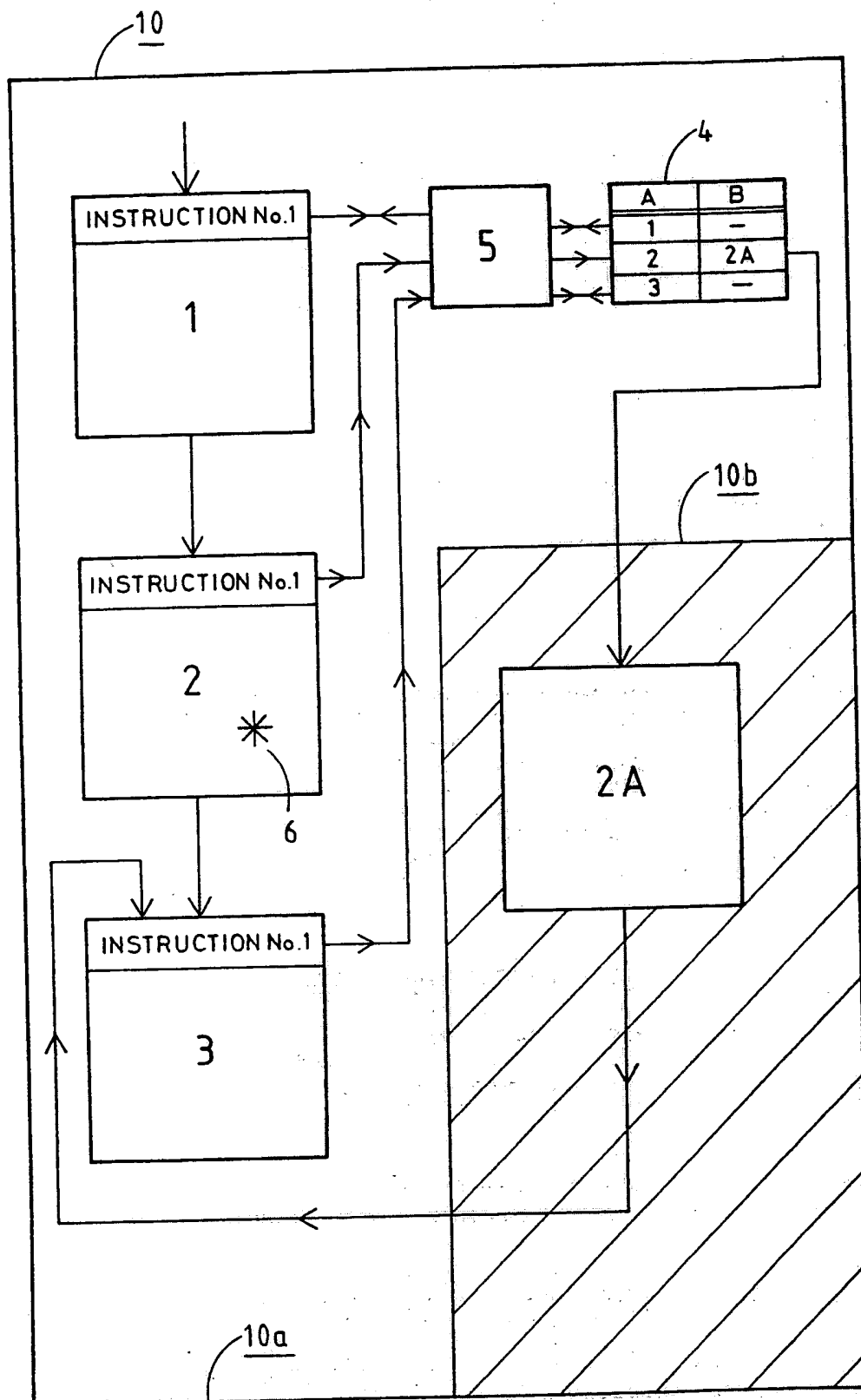


FIG. 1

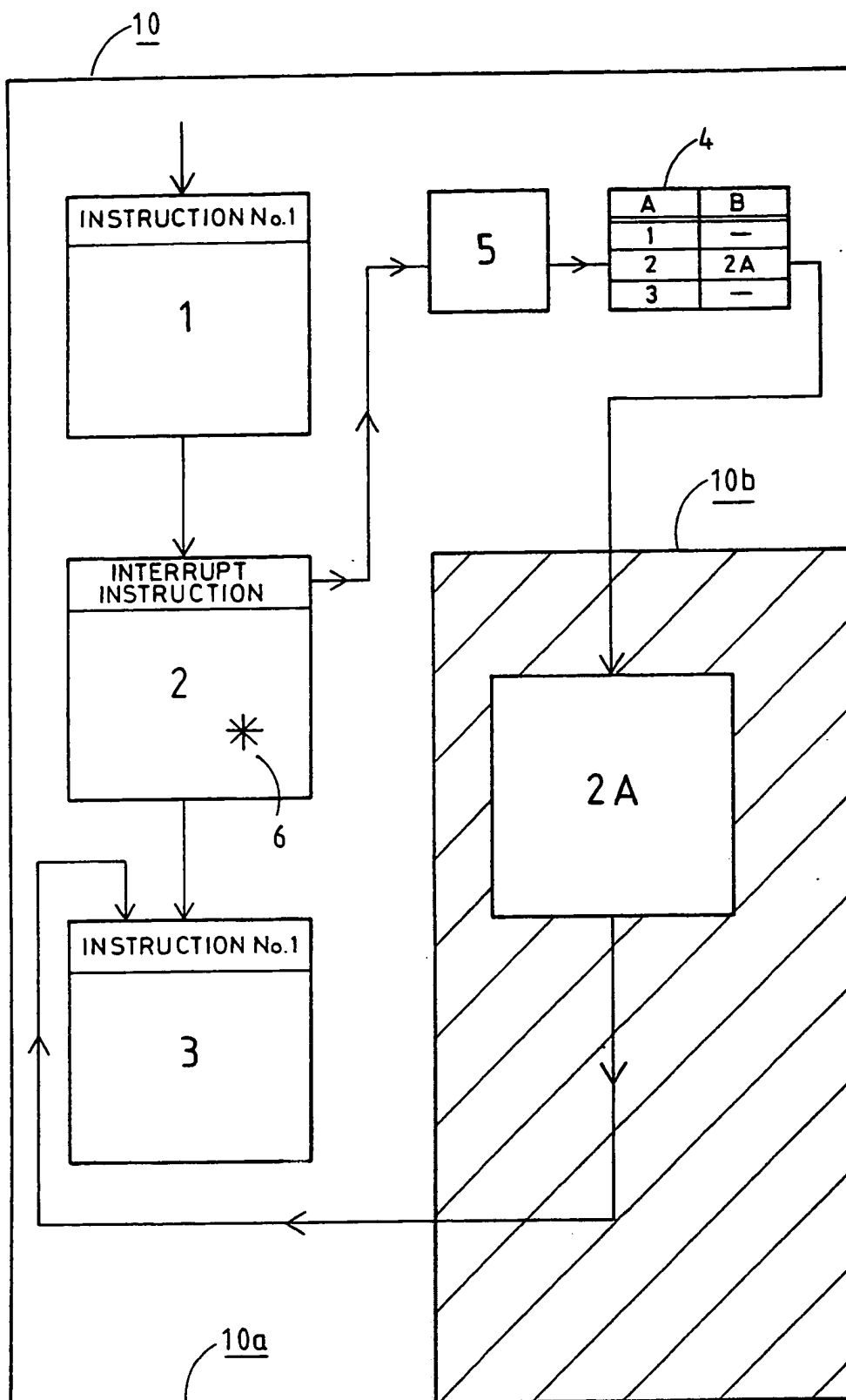


FIG. 2

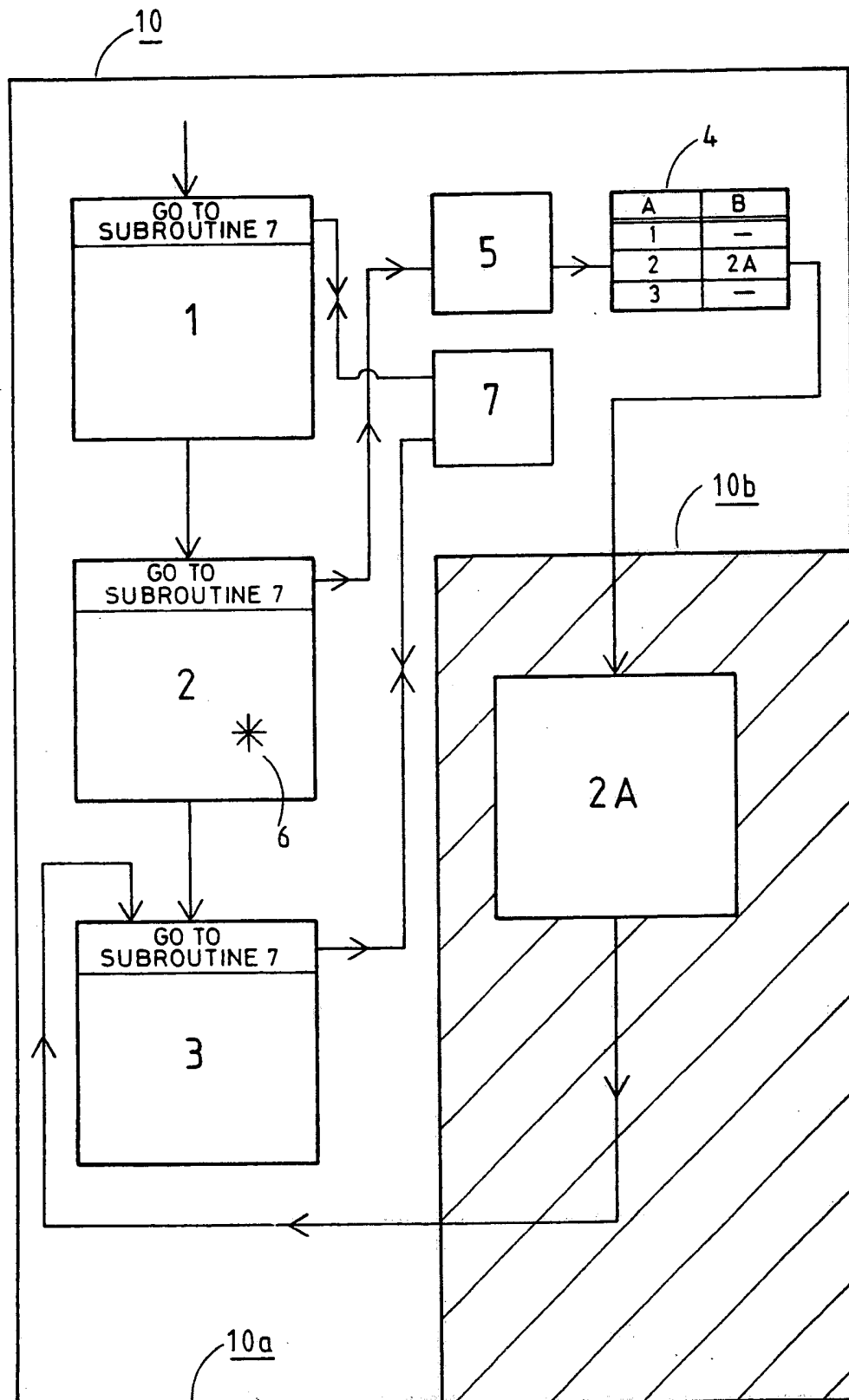


FIG. 3

-1-

## UPDATING PROM INFORMATION

This invention relates to a method of updating information stored in a programmable read only memory (PROM), and further relates to a PROM conditioned for use in such a method.

The term programmable read only memory (PROM) is used herein to refer to a memory device intended primarily for storing information to be read, but which can be programmed - at least once - by the user to alter the stored information. Thus, the term PROM specifically includes a one time programmable read only memory which, as the name implies, may be written on only once, as well as erasable devices known as EPROMs which may be written on several times.

It is known to store instructional information for execution by a computer (or microprocessor), especially information which is intended not to be changed or changed only infrequently, as so-called 'firmware' in a PROM. This contrasts with 'software' which is the type of instructional information which can be changed relatively easily, e.g. information stored on magnetic disk. PROM is a more secure vehicle for storing instructional information, for example an operating system, which the user is not ordinarily expected to change. However, problems arise when a programming error, or so-called bug, is found to exist in firmware because it has hitherto not always been easy or even possible to update information stored on PROM. EPROMs require physical access to the device itself to erase the information stored therein, whereas one time programmable read only memories cannot by definition have their information freely changed. It is not unusual, therefore, to discard the original PROM and to replace it with a new one, but this is obviously a costly and wasteful exercise.

According to a first aspect of the present invention there is provided a method of updating information stored in a programmable read only memory (PROM), the PROM having a portion comprising at least one pre-existing set of instructional information, an unprogrammed portion, and means for correlating information, the method including the steps of:

(a) storing in the unprogrammed portion a new set of instructional information, and

(b) entering into the correlating means information correlating the new set of instructional information with the pre-existing set,

an instruction being provided at the start of the pre-existing set of instructional information which causes the correlating means to be consulted before executing the remainder of the instructions in said pre-existing set whereby the new set of instructional information is executed instead of the pre-existing set when the correlating means comprises information correlating the new set with the pre-existing set of instructional information.

A method in accordance with the first aspect of the invention has the advantage that it enables instructional information, i.e. the firmware, stored in a PROM to be updated relatively easily without requiring physical access to the PROM device itself and without the need for discarding the original PROM. When a bug is found in a particular set of instructions, the whole of that set is re-written correctly in an empty part of the memory and thereafter the new set of instructions is executed instead of the original set. The original set of instructions is left unaltered, and is simply superseded by the new set.



In a particular embodiment, the method additionally includes the step of providing at the start of the pre-existing set of instructional information the instruction which causes the correlating means to be consulted. Suitably, this instruction may be provided by altering an existing instruction at the start of the pre-existing set of instructional information. The said existing instruction may either be a dummy instruction functionally independent of the remainder of the pre-existing set of instructional information, or it may form a functionally dependant part of the remainder of the pre-existing set of instructional information.

According to a further aspect of the invention there is provided a PROM conditioned for use in the method in accordance with the first aspect, the PROM comprising a pre-programmed set of instructional information, an unprogrammed portion into which the new set of instructional information may be written, means for correlating information about the preprogrammed set of instructional information with the new set, means for including an instruction at the start of the pre-programmed set of instructional information for causing the correlating means to be consulted before execution of the remainder of the instructions of the pre-programmed set, and a further set of instructional information for controlling the consultation of the correlating means whereby the new set of instructional information is executed instead of the pre-programmed set when the correlating means comprises information correlating the new set with the pre-programmed set of instructional information.

Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings in which:

Figure 1 is a schematic flowchart-style diagram illustrating a first method of updating information stored in a PROM in accordance with the present invention,

Figure 2 is a schematic flowchart-style diagram illustrating a second method in accordance with invention, and

Figure 3 is a schematic flowchart-style diagram illustrating a further method in accordance with the invention.

In Figure 1 there is shown a PROM or EPROM 10 having a portion 10a which is pre-programmed and a portion 10b (shown cross hatched) which is initially empty, i.e. unprogrammed. The pre-programmed portion 10a is shown as comprising three distinct sets of instructional information in the form of respective sub-routines 1,2,3 of a complete program. It will be evident to those skilled in the art that the complete program may perform any desired function, and as the specific function is not important to understanding the present invention no further details will be given here. Also, the skilled practitioner will realise that the complete program may - and usually will - comprise more (or less) than three subroutines.

Also, in the pre-programmed portion 10a there is provided a look-up table 4 shown schematically as comprising two columns A and B. In the first column A there is stored address information identifying the location in the PROM of each of the sub-routines 1 to 3. Column B of the look-up table 4 is originally empty. A special sub-routine 5 controls the periodic consultation of the look-up table 4 in the manner described in more detail below.

In the event of a programming error, or so-called bug, being found in one of the sub-routines 1-3, here represented by the

asterisk 6 in sub-routine 2, a new corrected version 2A of the whole of that sub-routine is written into the empty part 10b of the PROM in known manner. Also, the address of the new sub-routine is entered into the look-up table 4 in column B in correlated relationship to the address of the original sub-routine 2. The rest of column B is left empty.

As shown schematically, each of the subroutines 1-3 has a first instruction which calls up the sub-routine 5 for controlling the look-up table consultation procedure, as follows.

At the outset the first instruction of sub-routine 1 is executed which calls up sub-routine 5. This in turn causes the look-up table 4 to be consulted. On finding that there is no correlation information in the look-up table corresponding to sub-routine 1, sub-routine 5 proceeds back to sub-routine 1 for the whole of sub-routine 1 to be executed. Subsequently, the first instruction of sub-routine 2 is executed which again calls up sub-routine 5 causing the look-up table 4 to be consulted. This time the look-up table does contain information correlating the address of sub-routine 2 with that of newly-entered sub-routine 2A. Thus sub-routine 5 causes sub-routine 2A to be executed next instead of the remainder of original sub-routine 2. Hence the bug 6 in sub-routine 2 is bypassed. Sub-routine 2A proceeds to sub-routine 3. The first instruction of sub-routine 3 calls up the common sub-routine 5 which once again causes the look-up table to be consulted. As was the case for sub-routine 1, there is no correlation information held in the look-up table for sub-routine 3 and hence sub-routine 5 proceeds back to sub-routine 3 for the whole of sub-routine 3 to be executed and so terminate the whole program stored in the PROM.

In this example the common sub-routine 5 is executed at the start of each separate sub-routine 1-3, regardless of whether or not the sub-routine contains a bug.

Figure 2 shows a variation of the scheme described with reference to Figure 1. In this case the first instruction of each sub-routine originally forms a functionally dependant and integral part of the remainder of that sub-routine. In other words the first instruction of the sub-routines 1-3 does not ordinarily call up the look-up table consultation routine 5, which was the situation in the previous embodiment. However, if a sub-routine, such as sub-routine 2, does contain a bug 6, the first instruction of that sub-routine is converted to a particular code, specifically all zeros (e.g. 0000 0000 in 8-bit form) in the case of a one time programmable read only memory which interrupts sub-routine 2 and causes sub-routine 5 to be initiated. Sub-routine 5 operates exactly as described previously so that when the look-up table 4 is consulted and found to contain correlation information for sub-routine 2, namely the address of new sub-routine 2A, the new sub-routine is executed instead of original sub-routine 2. Thereafter, the sub-routine 3 is executed in its entirety.

In this example the look-up table consultation routine 5 is called up only at the start of a sub-routine containing a bug, i.e. in response to an interrupt command programmed specifically to bypass the defective sub-routine.

This scheme relies on the fact that it is possible (and relatively straightforward) to change the binary 'ones' of a PROM to binary 'zeros' although it is not possible to change the zeros to ones, except in the case of an electrically erasable programmable read only memory (EEPROM).

The scheme shown in Figure 3 is similar to that in Figure 2, except that the first instruction(s) [GO TO SUBROUTINE 7] of each sub-routine 1-3 ordinarily calls up an empty (or null) sub-routine 7 which simply returns to the original sub-routine for the remainder of that sub-routine to be executed in full. Here, the first instruction(s) of the sub-routines 1-3 are effectively dummy instructions. Again it is sub-routine 2 which contains the bug 6 and the first instruction of that sub-routine is converted to a new code [GO TO SUBROUTINE 5] by converting some or all of the existing binary ones of the original code to zeros, the new code initiating the look-up table consultation routine 5 instead of the empty sub-routine 7. Hence, newly written sub-routine 2A is executed instead of original sub-routine 2 and subsequently sub-routine 3 will be executed in its entirety, as before. This scheme also relies on the fact that it is possible (and relatively straightforward) to change the binary ones of a PROM to binary zeros. Alternatively, with an EEPROM the first instruction of a defective sub-routine may be changed to any suitable new code since not only can ones be changed selectively to zeros, but also the zeros can be changed selectively to ones.

In view of the foregoing description, it will be evident to a person skilled in the art that various modifications may be made within the scope of the present invention.

CLAIMS

1. A method of updating information stored in a programmable read only memory (PROM), the PROM having a portion comprising at least one pre-existing set of instructional information, an unprogrammed portion, and means for correlating information, the method including the steps of:

(a) storing in the unprogrammed portion a new set of instructional information, and

(b) entering into the correlating means information correlating the new set of instructional information with the pre-existing set,

an instruction being provided at the start of the pre-existing set of instructional information which causes the correlating means to be consulted before executing the remainder of the instructions in said pre-existing set whereby the new set of instructional information is executed instead of the pre-existing set when the correlating means comprises information correlating the new set with the pre-existing set of instructional information.

2. A method of updating information stored in a PROM as claimed in claim 1, further including the step of providing at the start of the pre-existing set of instructional information the instruction which causes the correlating means to be consulted.

3. A method of updating information stored in a PROM as claimed in claim 2, wherein the step of providing the instruction which causes the correlating means to be consulted comprises altering an existing instruction at the start of the pre-existing set of instructional information.

4. A method of updating information stored in a PROM substantially as herein described with reference to any of Figures 1 to 3 of the accompanying drawings.

5. A PROM conditioned for use in the method according to any one of claims 1 to 4, comprising a pre-programmed set of instructional information, an unprogrammed portion into which the new set of instructional information may be written, means for correlating information about the pre-programmed set of instructional information with the new set, means for including an instruction at the start of the pre-programmed set of instructional information for causing the correlating means to be consulted before execution of the remainder of the instructions of the pre-programmed set, and a further set of instructional information for controlling the consultation of the correlating means whereby the new set of instructional information is executed instead of the pre-programmed set when the correlating means comprises information correlating the new set with the pre-programmed set of instructional information.

6. A PROM as claimed in claim 4, wherein there is provided at the start of the pre-programmed set of instructional information a first instruction capable of being converted into the correlating means consultation instruction.

7. A PROM as claimed in claim 6, wherein the first instruction is a dummy instruction functionally independent of the remainder of the pre-programmed set of instructional information.

8. A PROM as claimed in claim 6, wherein the first instruction forms a functionally dependant part of the remainder of the pre-programmed set of instructional information.

9. A PROM conditioned for use in the method of any one of claims 1 to 4, substantially as herein described with reference to any of Figures 1 to 3 of the accompanying drawings.

DOCKET NO: GR98P5928P

SERIAL NO: 09/882,730

APPLICANT: Weber et al.

LERNER AND GREENBERG P.A.

P.O. BOX 2480

HOLLYWOOD, FLORIDA 33022

TEL. (954) 925-1100